Assessed Lab 1 (20%)

Due: 11:50am on 5 Nov 2018

Name: _____

User ID: _____

1 Rules

- **Do not turn over the page until asked to do so.**
- **This sheet must not leave the lab.**
- **You must work on your own.** Chat, email, message programs, and mobile phones must not be used during the assessed lab. If you talk to other students, or communicate by other means, you will receive a mark of 0.
- **You may not download Java code from any website except the module web page.** You can refer to code that you have written in previous lab classes. You may use any resources from the module web site (lecture slides, example code), from the Java API documentation, and you may refer to textbooks and lecture notes. You may use Google Translate if English is not your native language. **You may not refer to any other resources.**

  Module website: `http://staffwww.dcs.shef.ac.uk/people/H.Christensen/teaching/com6516/`
  Java API: `https://docs.oracle.com/javase/8/docs/api/`

- Upload your `*.java` files to MOLE using the assignments button on the menu bar. Do not submit any other files, such as `*.class` files.

  Your code must compile and run under Java 1.8, using `javac` and `java` on the command line. You can use the `sheffield` package already provided in this module and you may assume that it will be available to compile and run your code. You do not need to submit any of the `sheffield` files through MOLE.

- When you are finished, **you must hand in this sheet** with your name and CiCS user ID above. If you are satisfied with your work and submit before the deadline, you may leave the lab early without talking to any other students.
- You must stop working at the end of the lab session and submit your program code as described above. You should submit your code by 11:50; after this, MOLE will record the work as late and there will be a 10% penalty when it is marked. You will not be able to submit any code after 12:00.

  You can submit your work more than once; the last version submitted will be marked.

- If you are unable to upload your code to MOLE, please email your `*.java` files as attachments to `heidi.christensen@sheffield.ac.uk` with the subject line "COM6516: Assessed Lab 1".
- You may ask the demonstrators questions only if you are having technical problems with the lab machines or you do not understand what is being asked for in the assignment. You may not ask them questions about how to use particular Java constructs.

2 Task

Read all of the following instructions carefully before you start.

The programming task is to implement a simple tool for generating an walking plan for an old person to get them to be more active in their daily life. The tool should give you a different walking target (in meters) for each day and make sure you don't walk too much two days in a row.

Your program should start by asking for the name of the old person (the user). When they have typed in their name, the tool should then ask how old they are. You should personalise your question to them to include their name, e.g., "Hello Heidi, how old are you?"

The plan should cover 14 days. For each day in the plan, you should generate a random walking target between 100 and 2500 meters and in multiples of 10. That is, targets should have a value that is divisible by 10 (so, 120m, 490m and 2090m are all acceptable; 451m, 2011m are not) (hint: `java.util.Random`'s `nextInt(n)` returns $0 \leq x < n$ where x is the returned random number between zero and n).

Before printing your walking targets, you should check whether you have two *hard* days in a row in your plan (where we define a hard day as having a target > 1500 meters). If you have two hard days in a row, change the second day to 1000, so the user doesn't get too tired!

When you have made your plan, and made sure there are not too many hard days, you should print it out for the user. You should start by printing their name and age and then print the target for each in the plan. In the print out, you should indicate any hard days, for example by adding "(hard)" to that line.

At the end, you should print out the total number of meters the person will have walked in the plan as well as the average number of meters per day. You should calculate this value using a double and round it to the nearest integer (hint: use `Math.round(x)` found in `java.lang.Math`).

An example output is given here:

```
$ java GenerateWalkingPlan
What is your name? Heidi
Hello Heidi; how old are you? 100

Heidi (age=100) - this is your walking plan:
Day 1: walk 2170m <--- hard
Day 2: walk 1000m
Day 3: walk 940m
Day 4: walk 2190m <--- hard
Day 5: walk 1000m
Day 6: walk 290m
Day 7: walk 1150m
Day 8: walk 1820m <--- hard
Day 9: walk 1000m
Day 10: walk 20m
Day 11: walk 2100m <--- hard
Day 12: walk 1000m
Day 13: walk 1440m
Day 14: walk 1150m

Total number of meters walked = 17270
Average number of meters walked per day = 1233
```

You need to write a class called `GenerateWalkingPlan` (containing a `main` method), that asks the user their name and age, creates an instance of a `WalkingPlan` class and prints out the plan as described above. The `WalkingPlan` class should contain an array where the generated, random walking targets could be stored.

The `WalkingPlan` should contain a toPrint() method. You must decide what other useful attributes and methods each class should have.

Upload your `GenerateWalkingPlan.java`, and `WalkingPlan.java` files through MOLE. Your solution must compile and run under Java 1.8 on the command line using `javac GenerateWalkingPlan.java` and run using `java GenerateWalkingPlan`.

Remember that you need to create an instance of `Random`; the methods in the hints above are not static. You do not need to worry about seeding the random number generator or handling invalid input (assume the user will only enter a name and then a valid integer for the age).

3 Marking

Your submission will be marked out of 20, with up to 17 marks for code that compiles and produces correct output, and 3 marks for good programming style, including consistent indentation, sensible use of comments, and brace placement.

Marks and feedback will be returned to you through MOLE within 3 weeks.